Verifiable Credentials WG Miami 2023

Day 1: February 14, 2023
Chairs: Brent Zundel, Kristina Yasuda
Location: Miami (and the world wide web)

Welcome!

- Logistics
- W3C WG IPR Policy
- Agenda
- IRC and Scribes
- Status
- Timeline Reminder

Logistics

- **Zoom call:**
    - See https://mit.zoom.us/j/99464040866?pwd=QUpZSnRZU3FWMmRjcE9ZWWhQZ1ViUT09 for dial in information (member only link)
- **Meeting times**:
    - Tuesday Feb 14: 9:00-17:30 EST
    - Wednesday Feb 15: 9:00-14:00 EST
    - Friday Sep 16: 9:00-16:00 EST
- **VC WG Agenda**: https://tinyurl.com/vcwg-miami
- **Live slides**: https://tinyurl.com/vcwg-miami-slides (Google Slides)

W3C WG IPR Policy

- This group abides by the W3C patent policy https://www.w3.org/Consortium/Patent-Policy-20200915/
- Only people and companies listed at https://www.w3.org/groups/wg/vc/participants are allowed to make substantive contributions to the specs
- Code of Conduct https://www.w3.org/Consortium/cepc/

IRC and Scribes

- Meeting discussions will be documented
  - Text Chat: http://irc.w3.org/?channels=vcwg
  - IRC://irc.w3.org:6665/#vcwg
- Telecon info
  - https://www.w3.org/events/meetings/fa87947a-7cb6-4291-8c4f-34769b501551/20230214T090000#join

	Morning 1	M2	Afternoon 1	A2
Tues	Manu	Gabe	Will	Orie
Wed	Joe	Paul	David W	Mike Jones
Thurs	Mahmoud	Phil F	Phil F	Will

```
<JoeAndrieu> q+ to comment on biometrics
<brent> ack JoeAndrieu
<Zakim> JoeAndrieu, you wanted to comment on biometrics
```

Today’s agenda

9:00	Chairs Introduction and logistics	Chairs
9:30	Content Types	Orie
11:00	Coffee Break	
11:15	Content Types (vc-jws, vc-cose)	Orie
12:15	Lunch	
13:15	Holder Binding	Oliver
14:15	VC Extension points (evidence, statuslist, termsofuse, display, etc.)	Chairs
15:45	Coffee Break	
16:00	Terminology-related issues	Chairs
17:30	Dinner!	

VC WG Mission and Goals

- “The mission of the Verifiable Credentials Working Group is to make expressing, exchanging, and verifying credentials easier and more secure on the web.”

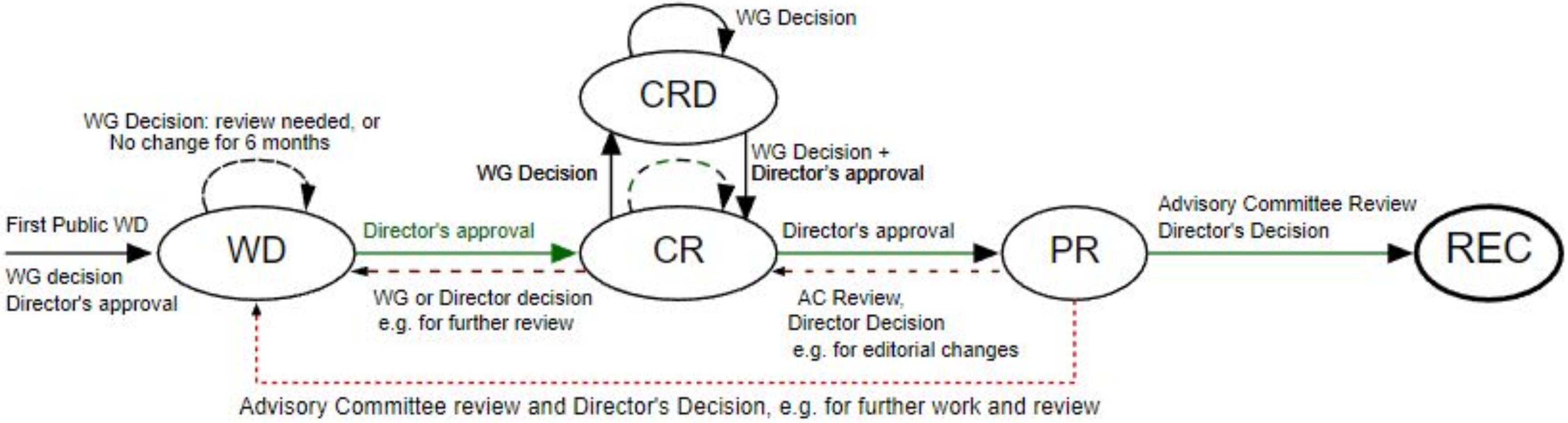
Charter Deliverables and Status

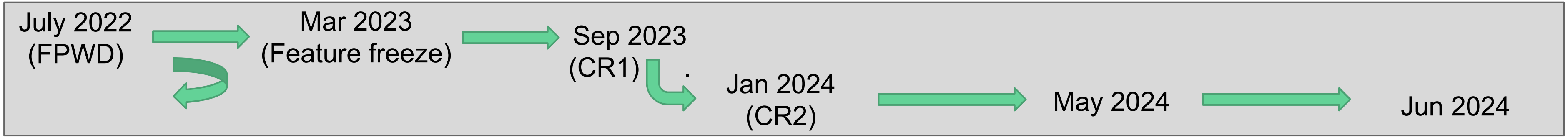
- **Verifiable Credentials Data Model (VCDM) 2.0**
- **VC Data Integrity 1.0**
- **VC JSON Web Token**
- **VC JSON Web Signature 2020**
- **VC Status List 2021**
- **VC EdDSA**

W3C Technical Report Process

- Working Draft (WD) - does not imply consensus
- Candidate Recommendation (CR)
    - Entry - to publish as CR, the document is expected to be feature complete, have had wide review, and must specify the implementation requirements needed to exit
    - Exit - to exit CR (and move to PR), the document must satisfy the stated implementation requirements; it must also not have made any substantive change not warned about upon entry
- Proposed Recommendation (PR)
    - Basically a one-month sanity check during which the AC is encouraged to have any final review and discussion, but if anything major happens it's a fail (requiring a move back to CR or earlier)
- Recommendation - Done
    - But errata are possible

Timing of our primary spec





July 2022
(FPWD)

Mar 2023
(Feature freeze)

Sep 2023
(CR1)

Jan 2024
(CR2)

May 2024

Jun 2024

https://www.w3.org/2021/Process-20211102/

Goals for this meeting

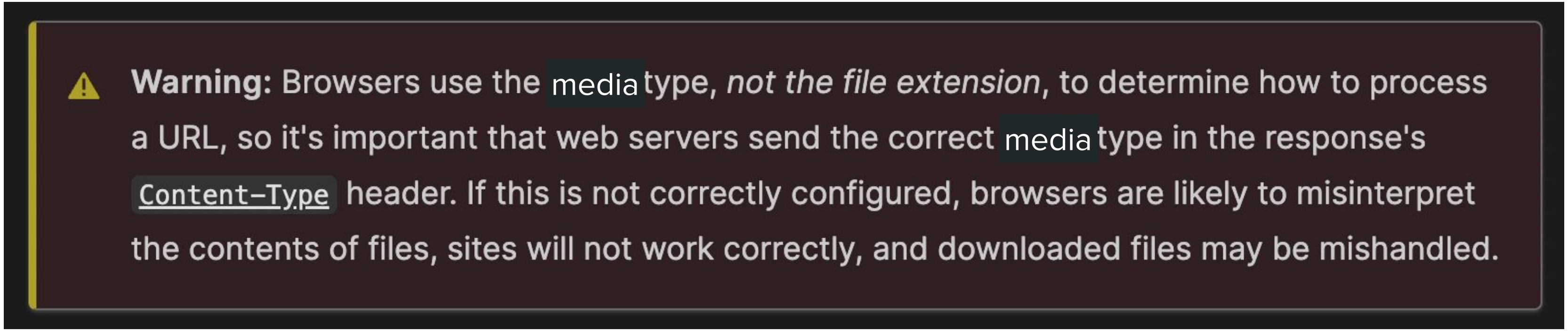
- Main Spec
  - Feature Freeze
- Other Normative Deliverables
  - A solid timeline for CR
- Non-normative deliverables
  - A full understanding of the set of work for each

Introductions

Content Types
(Orie, 60 min)

What is a content type / media type?

- en.wikipedia.org/wiki/Media_type
- developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP



⚠ **Warning:** Browsers use the media type, *not the file extension*, to determine how to process a URL, so it's important that web servers send the correct media type in the response's `Content-Type` header. If this is not correctly configured, browsers are likely to misinterpret the contents of files, sites will not work correctly, and downloaded files may be mishandled.

🎻🎵...Ominous foreboding music... & 🌶️ ... 🎻🎵

Where are media types registered?

- https://www.ietf.org/mailman/listinfo/media-types
- https://www.iana.org/assignments/media-types/media-types.xhtml
- W3C & IETF can request a registration...

Examples:

```
application/did+ld+json
application/credential+ld+json
```

How are media types used?

- JSON Web Signature — RFC7515#section-4.1.10

  The "`cty`" (content type) Header Parameter is used by JWS applications to declare the media type [IANA.MediaTypes] of the secured content (the payload).

- OAuth 2.0 — RFC6749#section-4.1.4

  The parameters are included in the entity-body of the HTTP response using the "`application/json`" media type as defined by [RFC4627].

- JSON-LD & Link Header — www.w3.org/TR/json-ld11

  Please note that JSON-LD documents served with the "`application/ld+json`" media type MUST have all context information, including references to external contexts, within the body of the document.

How are media types used in APIs?

- swagger.io/docs/specification/media-types

  Media type is a format of a request or response body data. Web service operations can accept and return data in different formats, the most common being JSON, XML and images. You specify the media type in request and response definitions.

- jsonapi.org/format/#media-type-parameters

  Note: A media type parameter is an extra piece of information that can accompany a media type. For example, in the header `` `Content-Type: text/html; charset="utf-8"` ``, the media type is text/html and charset is a parameter.

- developer.mozilla.org/en-US/docs/Web/API/Navigator/mediaCapabilities#examples

```
const mediaConfig = {
    type : 'record', // or 'transmission'
    video : {
        contentType : "video/webm;codecs=vp8.0", // valid content type
        width : 1920,     // width of the video
        height : 1080,    // height of the video
        bitrate : 120000, // number of bits used to encode 1s of video
        framerate : 48    // number of frames making up that 1s.
     }
};
```

```
// check support and performance
navigator.mediaCapabilities.encodingInfo(mediaConfig).then((result) => {
    console.log(`This configuration is ${result.supported ? '' : 'not '}supported,`);
    console.log(`${result.smooth ? '' : 'not '}smooth, and`);
    console.log(`${result.powerEfficient ? '' : 'not '}power efficient.`);
});
```

“application/credential+ld+json”

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2"
  ],
  "id": "http://example.edu/credentials/1872",
  "type": ["VerifiableCredential", "NewCredentialType"],
  "issuer": {
    "id": "did:example:123",
     "type": ["Organization", "OrganizationType"]
   },
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:456",
    "type": ["Person", "JobType"],
    "claimName": "claimValue"
  },
  “proof”: { … } // 🌶️ allowed? PR#1014
}
```

“application/verifiable-credential+ld+json”

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2"
  ],
  "id": "http://example.edu/credentials/1872",
  "type": ["VerifiableCredential", "NewCredentialType"],
  "issuer": {
    "id": "did:example:123",
     "type": ["Organization", "OrganizationType"]
   },
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:456",
    "type": ["Person", "JobType"],
    "claimName": "claimValue"
  },
    “proof”: { … } // 🌶️ required? PR#1014
}
```

“application/credential-claims-set-1.1+json”

```
{
  "sub": "did:example:ebfeb1f712ebc6f1c276e12ec21",
  "jti": "http://example.edu/credentials/3732",
  "iss": "did:example:123",
  "nbf": 1541493724,
  "iat": 1541493724,
  "exp": 1573029723,
  "nonce": "660!6345FSer",
  "vc": {
    "@context": [ // 🧊 :ice cube: Required.
        "https://www.w3.org/2018/credentials/v1",
        "https://www.w3.org/2018/credentials/examples/v1"
      ],
    "type": ["VerifiableCredential", "UniversityDegreeCredential"],
    "credentialSubject": {
      "degree": {
        "type": "BachelorDegree",
        "name": "Bachelor of Science and Arts"
      }
    }
  }
}
```

Break
(15 mins)

Content Types (vc-jws, vc-cose)
(Orie, 60 min)

How can we “secure” the media types defined in the core data model?

Define the new media types we want to secure.

Describe how to secure the defined media types in separate specs:

- https://github.com/w3c/vc-data-integrity
- https://github.com/w3c/vc-jwt

application/vc+ld+jwt (proposal)

```
{
  "kid": "https://example.edu/issuers/14#key-0",
  "alg": "ES256",
  "typ": "vc+ld+jwt" // 🌶️allowed? #51
   "cty": "credential+ld+json" // 🌶️ required?
}
```

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": [
    "VerifiableCredential",
    "UniversityDegreeCredential"
  ],
  "issuer": "https://example.edu/issuers/14",
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:123",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  },
“proof”: { … } // 🌶️ allowed?
PR#1014
}
```

application/vc+ld+cwt (proposal)

```
[
    / kid / 4: "https://example.edu/issuers/14#key-0"
    / alg / 1:-7, / ECDSA 256 /
    / typ / 61 ?? TBD ??? :  "vc+ld+cwt" // 🌶️allowed? #51
    / ctyp / 3: 0 "credential+ld+json"  // 🌶️required?
  ]
```

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "http://example.edu/credentials/3732",
  "type": [
    "VerifiableCredential",
    "UniversityDegreeCredential"
  ],
  "issuer": "https://example.edu/issuers/14",
  "issuanceDate": "2010-01-01T19:23:24Z",
  "credentialSubject": {
    "id": "did:example:123",
    "degree": {
      "type": "BachelorDegree",
      "name": "Bachelor of Science and Arts"
    }
  },
“proof”: { … } // 🌶️ allowed? PR#1014
}
```

Lunch
(50 mins — resume at 13:25 ET)

Holder Binding

(Oliver, 90 min)

Holder Binding

Oliver Terbu and David Chadwick

@W3C VCWG F2F Miami, Feb 14th, 2023

Problem Statement (Oliver/David)

- How can the verifier trust that the entity that presents a verifiable presentation is entitled to present the embedded VCs and the entity did not simply get a copy of these verifiable credentials?
  - Relates to relationships between presenter, issuer and subject, …
- How can the verifier trust that the entity, the one the issuer issued the verifiable credentials to, confirmed the verifiable presentation and the entity did not simply get a copy of the included verifiable credentials? (Oliver)
  - Relates to wallet security, consent, authentication (strong/multi-factor, certain level of assurance), selective disclosure, non-correlatable identifier, enabling pure online use cases, …

Assumptions (Oliver/David)

- Issuer should never know who the verifier is
- Issuer may attempt to control the use of the verifiable credential through the `termsOfUse` property,
  ... but binding the subject identifier to confirmation method(s) is a claim made by the issuer about the subject. (Oliver)

- **Issuers are trusted ➔ If not, then don't accept the VCs they issue**

  - Each issuer is making statements of facts (in its opinion) about the VC it is issuing.

  - The issuer MUST verify these statements before inserting them into the VC, otherwise it is not acting in a trustworthy manner. So, the issuer must be trusted by the verifier to make correct statements, or else it will not accept these VC.
    ➔ *also applies to binding/confirmation method(s) (Oliver)*

  - The issuer may insert `evidence` into the VC to tell the verifier which procedures it followed when asserting these facts ➔ important topic but not in scope of this session!

Confirmation Method

- A mechanism that produces a confirmation result which is endorsed by the issuer of a verifiable credential to verify that the intended bearer of the verifiable credential confirmed the verifiable presentation of that verifiable credential. A confirmation method is a claim made by the issuer as any other claim but with a specific semantic.
- Confirmation methods contain information on how to verify a confirmation result as well as can contain metadata about the confirmation process such as form factor of authenticators, assurance or confidence levels, etc.
- Verifiers that trust the issuer can also trust a conformation result by verifying a conformation result according to a specific confirmation type specification.

Consequences (Oliver/David)

- ANY property of the subject of a VC may be used by the verifier to determine if the presenter of the VP is the subject of a presented VC
    - *Also applies to claims about binding/confirmation method(s). (Oliver)*
    - *In certain cases, e.g., selective disclosure, non-correlatable identifier, support for ZKPs, there are not many useful claims that can be used for this. (Oliver)*
- The issuer does not need to tell the verifier what properties to use for verification, as this applies to all subject properties. The verifier decides which subject properties to use.
    - *If the verifier is interested in confirmation of the verifiable presentation, then it can get too complicated for the verifier to check everything themselves. (Oliver)*
    - *Since the verifier trusts the issuer, the verifier can also trust the binding/confirmation method endorsed by the issuer. (Oliver)*

Context Matters

- The term Holder Binding is misleading since the term depends strongly on the context the term is used in …
    - For example, the Dutch government definition of holder of an identity document is the person in whose name the identity document is issued and for whom it has been issued.
    - In VCDM, that person would be referred to as the subject (of the identity document), and the holder would be the entity that possesses it and can present it, which could be, but is not necessarily, its subject.

Holder Binding ➜ Identifier Binding

- Proposal based on RWOT Holder Binding Group
    - Paul Bastian, Rieks Joosten, Zaïda Rivai, Oliver Terbu
      Snorre Lothar von Gohren Edwin, Antonio Antonino
      Nikos Fotiou, Stephen Curran, and Ahamed Azeem
- RWOT paper can be found here
- Definition of Identifier Binding (from the RWOT paper): *“The process in which there is an identifier that a particular party has bound to some entity that it knows to exist, and has specified one or more means that other parties can use ~~to identify and/or authenticate~~ that entity. Such means are typically specified as part of a VC.”*

➜ However, although this slide deck uses examples from the paper, it deviates by saying “*to verify that entity confirmed the presentation*” where authentication is one aspect of the process and where confirmation methods are those “means”.

Use Case

- Bob offers a course "Making Logic Arguments Stick" in different scenarios ...
  - **Fully online course with no human teachers (FOCUS of this presentation)**
  - Fully online course where some classes are held by human teachers (see RWOT paper)
  - In-person classes with limited seats (see RWOT paper)
- Alice wants to enroll for Bob’s course in the scenarios above
- Bob requires the completion of another course “Second Order Logic” (SOL) before students can enroll for his course
- Ivan offers a course “Second Order Logic” to students
- Trevor wants to enroll Alice
- Mallory wants to enroll for Bob’s course without a “Second Order Logic” certificate by using Alice’s certificate.

Scenario 1 | Fully remote

- Alice receives a verifiable credential from Ivan upon completion of the course “Second Order Logic” (SOL).

- The verifiable credential does not contain any other claims than the credential subject passed the exam for the course SOL.

Scenario 1 | Fully remote

The verifiable credential looks like this:

```
...
"credentialSubject": {
    "id": "<some-uri>",
    "passedExam": "SOL"
}
...
```

Without additional information, the verifiable credential cannot be used by Alice to enrol for Bob's course in a secure way (online, in-person). Also Bob cannot verify that Alice confirmed any verifiable presentation that embeds the VC above.

Scenario 1 | Fully remote

We could add additional claims, so Bob could ask for an identity document that can be used to confirm Alice by comparing claims from the VC below against claims in the trusted identity document:

```
...
"credentialSubject": {
    "id": "<some-uri>",
    "passedExam": "SOL",
    "firstName": "Alice",
    "lastName": "Wonderland"
}
...
```

Implications

- Requires Alice to tell Ivan additional claims and provide evidence such as an identity document.
- If the identity document is not a digital credential, then it is very hard to use online.
- Alice cannot use the VC above in a pseudo/anonymous way anymore.
- When enrolling online there is no way for Bob to verify whether Alice confirmed the verifiable presentation herself. Note, Trevor could enrol Alice with the VC above for a in-person course and Alice would then show her identity document when going to the class but for fully remote courses this is very challenging.

Scenario 1 | Fully remote

Alice needs to present some cryptographic proof specifically for online use cases. Ivan could just include Alice's cryptographic key as a claim like this:

```
...
"credentialSubject": {
    "id": "<some-uri>", ; OR "did:..."
    "passedExam": "SOL",
    "publicKey": "..." ; OR "verificationMethod"
}
...
```

But this does not give the verifier enough guidance to verify the verifiable presentation since the verifier does not know the intention or further semantics of the `publicKey` claim. Note, if `verificationMethod` was used, it would be DID spec specific. A normative reference for such a mechanism is missing in the VCDM.

Scenario 1 | Fully remote

For that reason, we propose a new property `binding` (we can bikeshed the name another time, perhaps `confirmationMethod` is the better term):

```
...
"credentialSubject": {
    "id": "<some-uri>",
    "passedExam": "SOL",
    "binding": [{
        "type": "KeyConfirmationMethod2023",
        "publicKeyBase58": "did:key:z6...#key-1"
    }]
}
...
```

NOTE: DID/Key is not owned by issuer → issuer attested that claim after, for example, a DIDAuth challenge was verified.

This makes it clear and Alice can enroll for the Bob's course fully remotely. Mallory could also not just copy Alice's verifiable credential and enroll for Bob's course without passing the SOL exam first, since the verifiable credential is bound to a key that Alice possesses and that was endorsed by Ivan as the confirmation method for that VC. Only if Alice and Mallory would collude (friendly relay/fraud), Mallory would be able to confirm the verifiable presentation on behalf of Alice.

Scenario 1 | Fully remote (PROPOSAL)

Because confirmation methods can be quite different and a VC could have many confirmation methods endorsed by the VC issuer, we propose a new property where each confirmation method is described by its type and where each type is registered in a registry. The VCDM should also define 1-2 basic types such as `KeyConfirmationMethod2023.` For high assurance use cases it might not be even feasible to add all required claims to the VC (number too high, lack of standards, not possible to be verified by verifier etc.).

```
...
{
    "confirmationMethod": [
       { "type": "...", ... }, <more elements of array of confirmation-method-elements> ]
},
...
```

Scenario 1 | Other Examples

```
...
"credentialSubject": {
    "id": "<some-uri>",
    "passedExam": "SOL",
    "binding": [{
        "type": "DIDConfirmationMethod2023",
        "did": "did:key:z6..."
    },{
        "type": "KeyConfirmationMethod2023",
        "publicKeyBase58": "did:key:z6...#key-1",
        "loa": "high",  ; additional properties
                        ; allowed
        "some": "other" ; additional properties
                        ;allowed
        "realm": "eIDAS" ; additional properties
                         ; allowed
    },{
        "type": "AnonCredsLinkedSecret2023",
        "blindedLinkedSecret": "..."
    }]
}
...
```

```
...
"credentialSubject": {
    "id": "<some-uri>",
    "passedExam": "SOL",
    "binding": [{
        "type": "BiometricTemplate2023",
        "portrait": "data:..."
    }]
}
...
```

Scenario 1 | Linking Confirmation Methods

```
"credentialSubject": [ {
    "id": "uri:9021678535", //mandatory
    "binding": [ {
        "type": "secureWallet
                 RemoteBindingDIF",
        "walletName": "Example Wallet",
        //optional
        "walletVersion": "1.3.0",
        //optional
        "hardwarePublicKey":
           "did:jwk:123",
        //links and other formats
        // possible
        "holderAuthentication":
           ["FaceID", "PIN"]
    } ],
    "firstName": "Alice",
    "familyName": "Wonderland",
...
}
],
...
```

```
...

"credentialSubject": [ {
    "id": "uri:492754832663",
    "binding": [ {
        "type": "linkedBinding",
        "link": "uri:9021678535"
    } ],
    "hasPassedExam": "SOL"
}],
...
```

```
...
"credentialSubject": [ {
    "id": "uri:399912",
    "binding": [ {
        "type": "linkedCredential",
        "link": "uri:9021678535"
     } ],
     "isEnrolledFor": "MLAS-3"
}
],
...
```

Applications (non-exhaustive list)

- Where specific assurance or confidence levels are required to prevent from identity fraud
- Where common attacks such as replay, credential theft etc. has to be prevented where usually where the damage potential is high.
- Where pseudonymous verifiable credentials have to be preserved
- Where strong/multi-factor authentication is required and where it is hard for the verifier to verify the result from individual authenticators
- To support non-correlatable identifier and ZKPs
- To support fully online use cases where it is hard to provide additional evidence about the subject.

Applications (non-exhaustive list)

- Other Ecosystems using similar approaches
  - eIDAS 2.0 ARF [1] ➔ sole control / holder authentication
    - e.g. SIM eRegistration, Bank Account Opening, eDriving Licence, eGov Services, eSignature, ePrescription [2]
  - ISO 23220-4 Holder Confirmation Binding, ISO 18013-7 mDL
  - HyperLedger, Anoncreds
  - ICAO DTC ➔ Biometric Template sent upfront, so travelers can use pre-registered traveler programs and use automated border controls.

[1] https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework (2023)
[2] https://www.digital-identity-wallet.eu/ (2023)

Privacy Considerations

- Similar considerations as already described in the VCDM
- Selective disclosure of binding/confirmation methods and the use of non-correlatable identifier becomes even more important
- Some people might think that issuer endorsed binding or confirmation methods are too restrictive
  - But binding/confirmation methods are just one claim,
  - and verifiers can use other claims as well.

Break
(15 mins — resume 15:30 ET)

Alt Holder Binding

By omitting the concept of holder

David Chadwick, 13 February 2023

VC Roles

- There are four fixed roles (where the role occupant never changes during the life of a VC or VP)
  - For a VC there is the Issuer and the Issuee
  - For a VP there is the Presenter/Prover and the Verifier
- And one variable role (where the role occupant may change numerous times)
  - The Holder is whoever has a particular VC at any point in time.
- Conclusion. The role Holder lacks precision so we should use the terms Issuee and Presenter when we want to be more precise about the entity we are talking about

What is the Problem

that holder binding is trying to solve

1. ***How can the verifier tell that the presenter of a certain VP is entitled to present the embedded VCs and that he/she did not simply get a copy of these VCs from somewhere else or someone else***

2. If this is not the problem then what is it?

3. Noting always that the VC model is that the issuer should never know who the verifier is

4. But that the issuer may attempt to control the use of the VC through the TermsOfUse property

Issuers are Trusted

If not, then don’t accept the VCs they issue

1. Each issuer is making statements of facts (in its opinion) about the VC it is issuing.
2. The issuer MUST verify these statements before inserting them into the VC, otherwise it is not acting in a trustworthy manner. So, the issuer must be trusted by the verifier to make correct statements, or else it will not accept these VC.
3. The issuer may insert Evidence into the VC to tell the verifier which procedures it followed when asserting these facts
4. We already have one work item suggested to the CCG for KYC. See

https://docs.google.com/document/d/1htujrb-_1kh8tkV4MXYRmZ44m_D7yFrY09aFJkAz7io

Consequences

Of trusted issuer

1. ANY property of the subject of a VC may be used by the verifier to determine if the presenter of the VP is the subject of a presented VC

2. The issuer does not need to tell the verifier what properties to use for verification, as this applies to all subject properties. The verifier decides which subject properties to use.

What if the VC is not issued to the Subject?

Then it is issued to someone else - the issuee

1. When the issuee is not the subject, the issuer MAY insert an issuee property into the VC detailing who the issuee is. This again is a statement of fact by the issuer.

2. The issuee should be formatted as a JSON object (in the same way as issuer and subject currently are) so that it may contain any properties that can be used to identify the issuee, for example, a public key, DID, a name and address or biometric photo etc

3. The verifier will decide which of the issuee properties to use to verify that the presenter of the VP is the issuee of an embedded VC

Hypothesis. Property X

Can be used to solve the initial problem formulation

1. It is incumbent on the presenter to aid the verifier in determining that it is the rightful possessor of the embedded VCs by completing property X (if it wishes its VP to be accepted by the verifier)

2. Property X provides hints to the verifier to determine that the presenter is a valid possessor of the VCs that it is presenting

3. The verifier can ignore property X, or check that the statements made in property X are true or false.

4. Either way, the verifier decides whether to accept the VP or not, and property X may be used as an aid in this.

Property X in JSON

Inserted by the Presenter into the VP it signs

```
{
        "x": [{
                "credentialId": "id of an embedded VC",
                "type": "a URI",
                "other optional properties": "as defined by the type"
        }]
}
```

Some identified Types of X

Bearer

```
{
    "x": [{
        "credentialId": "id of an embedded VC",
        "type": "Bearer",
    }]
}
```

Semantics: the presenter is asserting to the verifier that the identified VC may be held by anyone and that proof of possession is not required.

Some identified Types of X

Subject

```
{
    "x": [{
        "credentialId": "id of an embedded VC",
        "type": "Subject",
    }]
}
```

Semantics: the presenter is asserting to the verifier that he/she is the subject of the identified VC. The verifier may chose any of the subject properties to verify this.

Some identified Types of X

Issuee

```
{
        "x": [{
                "credentialId": "id of an embedded VC",
                "type": "Issuee",
        }]
}
```

Semantics: the presenter is asserting to the verifier that he/she is the issuee of the identified VC. The verifier may chose any of the issuee properties to verify this.

Some identified Types of X

Related

```
{
        "x": [{
                "credentialId": "id of an embedded VC",
                "type": "Related",
        }]
}
```

Semantics: the presenter is asserting to the verifier that he/she is related to the subject of the identified VC. The verifier needs to consult the accompanying Relationship VC to determine what this relationship is.

Some identified Types of X

Relationship

```
{
        "x": [{
                "credentialId": "id of an embedded VC",
                "type": "Relationship",
        }]
}
```

Semantics: the presenter is asserting to the verifier that the identified VC was issued to itself (i.e. it is the issuee) and that this VC describes the relationship that the presenter has to the subject of this VC (who is also the subject of the VC identified in the Related array element)

Relationship VC

Just an example of what we might define e.g. for a parent child relationship

```
{       "@context": ["https://www.w3.org/2023/credentials/v2"],
        "id": "http://example.edu/credentials/58473",
        "type": ["VerifiableCredential", "Relationship"],
        "issuer": “https://example.edu/issuers/565049",
        "issuanceDate": "2020-01-01T00:00:00Z",
        "credentialSubject": {"name": "Name of Child",
                "DoB": "2020-01-01T00:00:00Z",
                "photo": "base64 image of child"},
        "issuee": {"id": “some DID or key ID",
                 "name": "Name of Relative",
                "DoB": "1980-01-01T00:00:00Z",
                "photo": "base64 image of relative"},
        "relationship": "parent|father|mother|brother|sister etc"
}
```

Some identified Types of X

Delegator

```
{
        "x": [{
                "credentialId": "id of an embedded VC",
                "type": “Delegator",
        }]
}
```

Semantics: the presenter is asserting that they have been delegated by the subject of the identified VC to act on that subject’s behalf. The verifier will need to consult a Delegation VC (or other mechanism) in order to determine what this delegation is

Some identified Types of X

Delegation

```
{
        "x": [{
                "credentialId": "id of an embedded VC",
                "type": “Delegation",
        }]
}
```

Semantics: the presenter is asserting that the identified VC has been issued to the presenter (i.e. the subject of the identified VC is the presenter) by the delegator and that this Delegation VC describes the delegation that has been given to the presenter by the delegator. (The delegator is also the subject of the VC identified in the Delegator array element.)

Delegation VC

Just an example of what we might define

```
{       "@context": ["https://www.w3.org/2023/credentials/v2"],
        "id": "http://example.edu/credentials/123",
        "type": ["VerifiableCredential", "Delegation"],
        "issuer": “subject.ID from delegator's VC",
        "issuanceDate": "2020-01-01T00:00:00Z",
        "credentialSubject": {"id": "key of delegate",
                "name": "Name of delegate",
                "DoB": "2020-01-01T00:00:00Z",
                "photo": "base64 image of delegate"},
        ""delegation": [{
                "resource": "URL of resource",
                "action": "read | write | enrol etc.”]}
}
```

Use Case Examples

Use of a Child’s Passport

- The child presents their own passport

```
{
        "x": [{
                "credentialId": "id of child's passport VC",
                "type": "Subject"
        }]
}
```

Use Case Examples

Use of a Child’s Passport

- A relative presents the child’s passport (along with their own)

```
{
	"x": [{
		"credentialId": "id of relative's passport VC",
		"type": "Subject"
	}, {
		"credentialId": "id of child's passport VC",
		"type": "Related"
	}, {
		"credentialId": “id of Relationship VC",
		"type": "Relationship",
	}]
}
```

- Verifier checks that the presenter is the subject of the Subject VC, then
- that the presenter is the issuee of the Relationship VC, then
- that the subject of the Relationship VC is the subject of the Related VC

Use Case Examples

Use of a Company VC

- Companies House issues VCs to companies containing details of the company: name, number, directors, secretary etc. If issued to a director or secretary, then they can present it to a verifier using

```
{
        "x": [{
                "credentialId": "id of company’s VC",
                "type": "Subject"
        }]
}
```

Use Case Examples

Use of a Company VC

- If an authorised person asks Companies House for a copy of the company’s VC it places the identity of the authorised person in the issuee property. The authorised person then presents this using

```
{
        "x": [{
                "credentialId": "id of company’s VC",
                "type": “Issuee"
        }]
}
```

Use Case Examples

Use of a Company VC

- If Companies House is willing to issue a copy of the company’s VC to anyone (i.e. the public) then it does not record anything special in the VC. The person holding the VC may present it using

```
{
        "x": [{
                "credentialId": "id of company’s VC",
                "type": “Bearer”
        }]
}
```

Use Case Examples

Alice-Bob-Trevor example

- Alice has a VC with her key ID to say she has passed the SOL exam. (Alice is anonymous in this case)
- To enrol on Bob’s course, Bob allows anyone to enrol anyone providing that the enrollee has passed the SOL exam. Without X controls, Alice, Trevor and Mallory may all enrol Alice if they have a copy of Alice’s (anonymous) SOL exam VC.
- To prevent this, Bob requires the applicant (Trevor) to have been delegated permission by the enrollee (Alice), and to include the following X in the presented VP

```
{
        "x": [{
                "credentialId": "id of Trevor’s VC",
                "type": "Subject"
        }, {
                "credentialId": "id of Alice’s SOL VC",
                "type": "Delegator"
        }, {
                "credentialId": “id of Delegation VC",
                "type": "Delegation",
        }]
}
```

- Verifier checks that the presenter (Trevor) is the subject of the Subject VC, then
- that the presenter is the subject of the Delegation VC, then
- that the issuer of the Delegation VC is also the subject of the Delegator VC
- that the Delegation VC gives permission to Enrol on the Course URL and
- that the Delegator has passed the SOL exam

VC Extension points (Chairs, 90 min)

Extension Points

- Dereferencable URLs — what do they dereference to?
    - Id, type, issuer, holder
- What can we normatively describe (or point to) for the following?
    - status — https://github.com/w3c/vc-status-list-2021
    - credentialSchema — https://github.com/w3c/vc-data-model/labels/schema
    - refreshService — https://github.com/w3c/vc-data-model/labels/refresh
    - termsOfUse — https://github.com/w3c/vc-data-model/issues/1010
    - evidence — https://github.com/w3c/vc-data-model/labels/evidence
    - render —

Break
(15 mins)

Terminology
(Chairs, 90 min)

End of Day 1

Verifiable Credentials WG Miami 2023

Day 2: February 15, 2023
Chairs: Kristina Yasuda, Brent Zundel
Location: Miami (and cyberspace)

Today’s agenda

9:00	Setting expectations for day 2	Chairs
9:15	vc-data-integrity (Security Vocab, etc)	Manu
10:15	Coffee Break	
10:30	vc-data-integrity (Security Vocab, etc)	Manu
11:00	vc-jwt	Orie
12:00	Lunch	
13:00	vc-jwt	Orie
13:30	(buffer time) Issue Processing	Chairs
14:30	Coffee Break	
14:45	Issue Processing	Chairs
16:00	activity	

Data Integrity
(Manu, 60 mins)

2023-02 - W3C VCWG F2F
VC Data Integrity Update

Update on status, important decisions, roadmap

Roadmap Status Update

Accomplishments since WG start:

- Adoption of VC Data Integrity
- Security Vocabulary cleanup
- FPWD Publication of VC Data Integrity
- Adoption of JsonWebSignature2020
- Adoption of EdDSA Cryptosuite
- FPWD of vc-jws-2020

To do:

- FPWD of EdDSA Cryptosuite
- ECDSA Cryptosuite?
- Complete test suites
- Lots of issue processing
- Snapshottable Candidate Recommendations?

Interoperability Overview

CHAPI and VC-API

81 Combinations Demonstrated

17 different Issuers

14 PlugFest participants

3 from the broader VC-API community

8 different wallets

5 web wallets

3 native mobile apps

ASU Arizona State University

ACREDITTA

API CATALOG

DANUBE TECH GMBH

DCC DIGITAL CREDENTIALS CONSORTIUM

Gobekli

INSTRUCTURE

LEARNING ECONOMY

Spruce

FUIX LABS.

PROOFSPACE

RANDA

participate.

Mavennet

veres

Snapbrillia beta

mesur.io

SmartResume

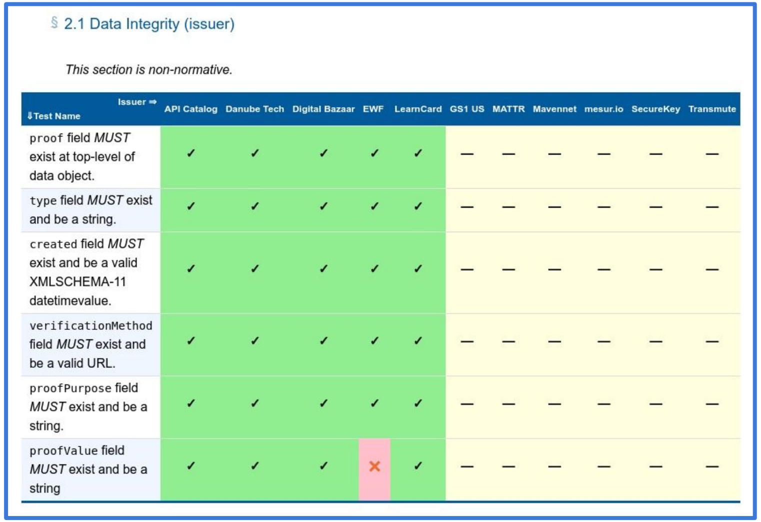
Status of Test Suites

We have preliminary test suites and enough implementations for:

- VC Data Integrity
- EdDSA Cryptosuite
- [JsonWebSignature2020]

We still need to:

- ECDSA Cryptosuite tests?
- Align test suites w/ latest spec text
- Get more implementers



§ 2.1 Data Integrity (issuer)

This section is non-normative.

Issuer ⇒ ⇓Test Name	API Catalog	Danube Tech	Digital Bazaar	EWF	LearnCard	GS1 US	MATTR	Mavennet	mesur.io	SecureKey	Transmute
`proof` field *MUST* exist at top-level of data object.	✓	✓	✓	✓	✓	—	—	—	—	—	—
`type` field *MUST* exist and be a string.	✓	✓	✓	✓	✓	—	—	—	—	—	—
`created` field *MUST* exist and be a valid XMLSCHEMA-11 datetimevalue.	✓	✓	✓	✓	✓	—	—	—	—	—	—
`verificationMethod` field *MUST* exist and be a valid URL.	✓	✓	✓	✓	✓	—	—	—	—	—	—
`proofPurpose` field *MUST* exist and be a string.	✓	✓	✓	✓	✓	—	—	—	—	—	—
`proofValue` field *MUST* exist and be a string	✓	✓	✓	✕	✓	—	—	—	—	—	—

Remaining important decisions

- Should all/certain cryptosuites support JSON Canonicalization Scheme (JCS) as well as the 2015 Universal RDF Dataset Canonicalization Algorithm? #25
- Should chains of proofs be expressed as a new type `ChainedDataIntegrityProof`, or an array of proofs via `proofChain`? #26
- Should `domain` be a string or an array? #29
- Readability of General Algorithms vs. Cryptosuite-specific algorithms

Potential Changes to JWS 2020

- Use JCS instead of URDNA?
  - Data Model is JSON not RDF… is JCS enough for VCs?
- How can this suite be different from the other suites in a way that justifies its existence?

Break
(15 mins)

Data Integrity
(Manu, 30 mins)

2023 VC Data Integrity Roadmap

Proposed roadmap:

- February: FPWD of EdDSA Cryptosuite
- March/April: ECDSA Cryptosuite adoption?
- May: Complete test suites
- June: Snapshottable Candidate Recommendations? Candidate Rec for URDNA2015?
- Aug: Candidate Recommendation Snapshot(s) 2
- Nov: Candidate Recommendation Snapshot(s) 3

VC-JWT
(Orie, 60 mins)

Registered Claim Names

The following Claim Names are registered in the IANA "JSON Web Token Claims" registry established by Section 10.1. None of the claims defined below are intended to be mandatory to use or implement in all cases, but rather they provide a starting point for a set of useful, interoperable claims. Applications using JWTs should define which specific claims they use and when they are required or optional. All the names are short because a core goal of JWTs is for the representation to be compact.

iss, sub, aud, exp, nbf, iat, jti, vp, vc

Public Claim Names

Claim Names can be defined at will by those using JWTs. However, in order to prevent collisions, any new Claim Name should either be registered in the IANA "JSON Web Token Claims" registry established by Section 10.1 or be a Public Name: a value that contains a Collision-Resistant Name. In each case, the definer of the name or value needs to take reasonable precautions to make sure they are in control of the part of the namespace they use to define the Claim Name.

urn:example:claim

Private Claim Names

A producer and consumer of a JWT MAY agree to use Claim Names that are Private Names: names that are not Registered Claim Names (Section 4.1) or Public Claim Names (Section 4.2). Unlike Public Claim Names, Private Claim Names are subject to collision and should be used with caution.

organization_id,

“application/verifiable-credential+jwt”

```
{
  "kid": "https://example.edu/issuers/14#key-0",
  "alg": "ES256",
  "typ": "vc+jwt" //shortened in #50
  "cty": "credential-claims-set+json"
}
```

```
{
 "@context": [ // 🧊allowed ... and NOT required.
     "https://www.w3.org/2018/credentials/v1",
     "https://www.w3.org/2018/credentials/examples/v1"
   ],
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022,
  "urn:example:claim": true
}
```

🧊 **#44 ...merged february 10th**

Lunch
(60 mins)

VC-JWT
(Orie, 30 mins)

SD-JWT VC?

https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt

```
{
  "alg": "RS256",
  "kid": "cAEIUqJ0cmLzD1kzGzheiBag0YRAzVdlfxN280NgHaA",
  "typ": "vc+sd+jwt", // 🌡 ???
  "cty": "credential+sd+json", // 🌡 ???
}
```

```
{
  "_sd": [
    "NYCoSRKEYwXdpe5yduJXCxxhynEU8z-b4TyNiap77UY",
    "SY8n2BbkX9lrY3exHlSwPRFXoD09GF8a9CPO-G8j208",
    "TPsGNPYA46wmBxfv2znOJhfdoN5Y1GkezbpaGZCT1ac",
    "ZkSJxxeGluIdYBb7CqkZbJVm0w2V5UrReNTzAQCYBjw",
    "l9qIJ9JTQwLG7OLEICTFBVxmArw8Pjy65dD6mtQVG5c",
    "o1SAsJ33YMioO9pX5VeAM1lxuHF6hZW2kGdkKKBnVlo",
    "qqvcqnczAMgYx7EykI6wwtspyvyvK790ge7MBbQ-Nus"
  ],
  "iss": "https://example.com/issuer",
  "iat": 1516239022,
  "exp": 1516247022,
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "RSA",
      "n": "pm4bOHB...",
      "e": "AQAB"
    }
  }
}
```

SD-JWT VP?

https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt

```
{
 "alg": "RS256",
 "kid": "cAEIUqJ0cm…",
 "typ": "vp+sd+jwt", // 🌡️ ???
 "cty": "presentation+sd+json", // 🌡️ ???
}
```

```
{
  "iss": "https://holder.example.com",
  "sub": "did:example:123",
  "aud": "https://verifier.example.com",
  "exp": 1590000000,
  "iat": 1580000000,
  "nbf": 1580000000,
  "jti": "urn:uuid:12345678-1234-1234-1234-123456789012",
  "_sd_jwt": "eyJhbGci...emhlaUJhZzBZ~eyJhb...dYALCGg~"
}
```

IETF JOSE WG is Back

We asked IETF to work on standardizing JWP.
CFRG is working on BLS 12-381 which some JWP plan use.

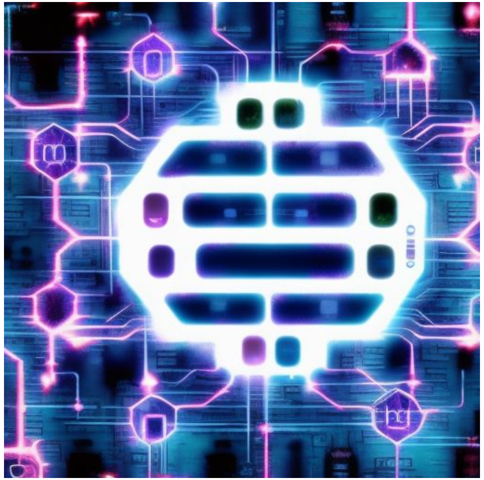
JWP

https://github.com/json-web-proofs/json-web-proofs

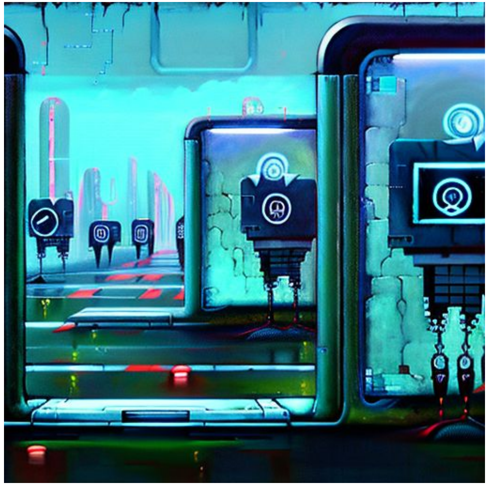
```
{
  "kid": "Hjfcpyj..",
  "alg": "BBS",
  "typ": "vp+jwp", // 🌡️ ???
  "cty": "credential-claims-set+json", //
🌡️ ???
  "claims": [
    "iat",
    "exp",
    "family_name",
    "given_name",
    "email"
  ]
}
```

```
{
  "payloads": [
    null,
    "IkpheSI",
    null,
    "NDI"
  ],
  "issuer": "eyJ...",
  "proof": "LJM...",
  "presentation": "eyJub25jZSI6InVURUIzNzF.."
}
```

Woman Savig



SAVING
710



Issue Processing (Chairs 60 mins)

Break
(15 minutes)

Issue Processing
(Chairs 75 mins)

Activity

Bayside Marketplace

401 Biscayne Blvd, Miami, FL 33132,

USA

Be there by 4:30pm

End of Day 2

Verifiable Credentials WG
Miami 2023

Day 3: February 16, 2023
Chairs: Kristina Yasuda, Brent Zundel
Location: Miami (and your house)

Today’s agenda

9:00	Setting expectations for day 3	Chairs
9:15	@context optional	Chairs/Gabe
10:15	Coffee Break	
10:30	@context optional	Chairs
12:30	Lunch	
13:30	Industry News (US, Canada, EU, LATAM, etc.)	Mike P
14:30	Issue triaging	Chairs/Editors
15:30	Coffee Break	
15:45	Issue triaging	Chairs/Editors
15:30	Deliverables	Chairs

@context
(Gabe and Chairs, 60 mins)

@context optional

Miami Edition

30+

participants

127

days (prime number!)

290

comments

The Key Question

Is the VCDM a JSON-LD data model?
Yes? There must be a @context.
No? Let’s make it optional.

Another Key Question

What type of interoperability are we aiming to support?

Last Key Question

Does the future of VCs look brighter
if we can compromise?

01

Background

Background

- We’ve had multiple special topic calls, months of discussions, and attempts at compromise
- With VCDM v2 we have the ability to break backwards compatibility and make @context optional
- In straw polls, the group seems consistently split
- There have been a few concrete proposals with concessions made
- Likely the split is due to two worldviews on “what a credential is,” how it should be extended, and what interoperability means

Arguments: Pro

- There is a concept of a VC that exists whether or not a @context is present.
- There are multiple meanings of interop. Semantic interop is not wanted/needed for some of them.
- Interoping with *all* verifiable credentials is a questionable requirement – let's interop with credentials that are worth interoping with.
  - Force interop vs enable interop
- There's a high barrier to understanding JSON-LD; not friendly for devs; confusing to many
- There are already credentials (yes, not spec complaint) that do not use LD properly, and don't wish to. This is useful to no one.
- Extensibility can work with JSON Schema, registries, and other mechanisms whether normative or non-normative.
- Other representations (e.g., ACDCs, ZKPs, CBOR, YAML, …) do not *always* have a need for LD
- Let's not preempt the interoperability we need. Let's work with the market and use cases we have now, which say LD is overkill for many applications of VCs.
- Coupling security with semantics is a confusing and possibly dangerous practice!
- VCs aren't JWTs. Lots of meaning and market use attached to them past @context

Arguments: Con

- Not ideal for interop (more than one way to do things). Would need to re-define what interop means.
- Can't reach consensus → we'll end up back where we are now. Has been discussed many times over the years.
- Why not just use existing standards for signing JSON or CBOR if the data model does not include semantics?
- Interop can be achieved by LD processors in other ways, if they care to do so.
- Was tried in the DID WG. Didn't go well.
- Adding @vocab to the base context solved most of the concerns. No longer needed to be optional.
- How would extensibility work without an open world model?
- There is a large burden on verifiers, we should not increase it with more divergent options.
- This comment (importance of the graph data model, etc.)
- Extensibility is tricky and LD is a decentralized extensibility mechanism that works.
- Registries are hard to maintain and don't work well for extensibility. See the DID WG.

02

Solutions?

@vocab in the credentials/v2 @context

- Merged in #1001, #953,
- Provides a “default namespace” for terms not defined by a context.
- Lowers the barrier for interop – pretty much get it by default
- **tl;dr**: context is not optional but you don’t need to ever touch it if you don’t care to

Media Types

- @context is required in media types that support linked data interoperability (ld+json)
- Not all media types support linked data interoperability
- The core data model remains JSON-LD, though not all representations of VCs utilize JSON-LD (they don't include a @context)
- As per VC-JWT PR #44 there is a media type (typ) `verifiable-credential+jwt` with optional content type (cty) `credential-claims-set+json` that does not require an @context property
- **Media types allow us to define representations that may or may not use @context!**

Layering

- Issue #982 from Sam Smith
- Let’s reconsider the spec in layers that allows us to gain flexibility and a better understanding of the concerns and problems each layer is attempting to address

Layers

- Authentication Layer and Authorization Sub-layer
- Presentation layer

Other Takes

- Credential → Credential Metadata → Proof Metadata → Proof
- Have semantics be an optional layer

Transformation

- As discussed yesterday
- Have a transformation context that can be used when going from vc-jwt to other credential formats

03

The Options

Proposals

Option 1

The @vocab compromise is sufficient

Option 2

Media types = freedom

Op 3

l

W r la

Option 4

Transformation

Proposals

Option 1: @context required

The @vocab compromise is sufficient

Option 2: @context optional

Media types = freedom with possible Transformation

Brent's 6 Questions

1. Is the VC Data Model strictly an RDF Data Model?
2. Beyond 'semantic interoperability', what does `@context` provide?
3. If we keep a single base media type of `credential+ld+json`, what can you not do?
4. Must all VCWG-registered media types include ld?
5. Are there constraints that could be added to the media types option that would make it palatable?
6. If `@context` is made optional, what can you not do?

Proposal

1. @context is required (MUST) in the base media type; SHOULD in other media types
2. Base media type is `credential+ld+json`
3. Utilize parameterized media types to create transformations to/from the base media type; the transformations SHOULD be lossless
4. add "Verifiable credentials define terms in a JSON-LD context at https://www.w3.org/ns/credentials/v2. Implementers SHOULD include the verifiable credential context in their object definitions. Implementers MAY include additional context as appropriate."
5. all representations of the VCDM MUST have a property that conveys versioning information

Proposal

POLL passed conditional to item 3.

1. The base media type for the VCDM is credential+ld+json.
2. @context is required (MUST) in the base media type; other media types MAY choose to include @context.

Trying to refine:

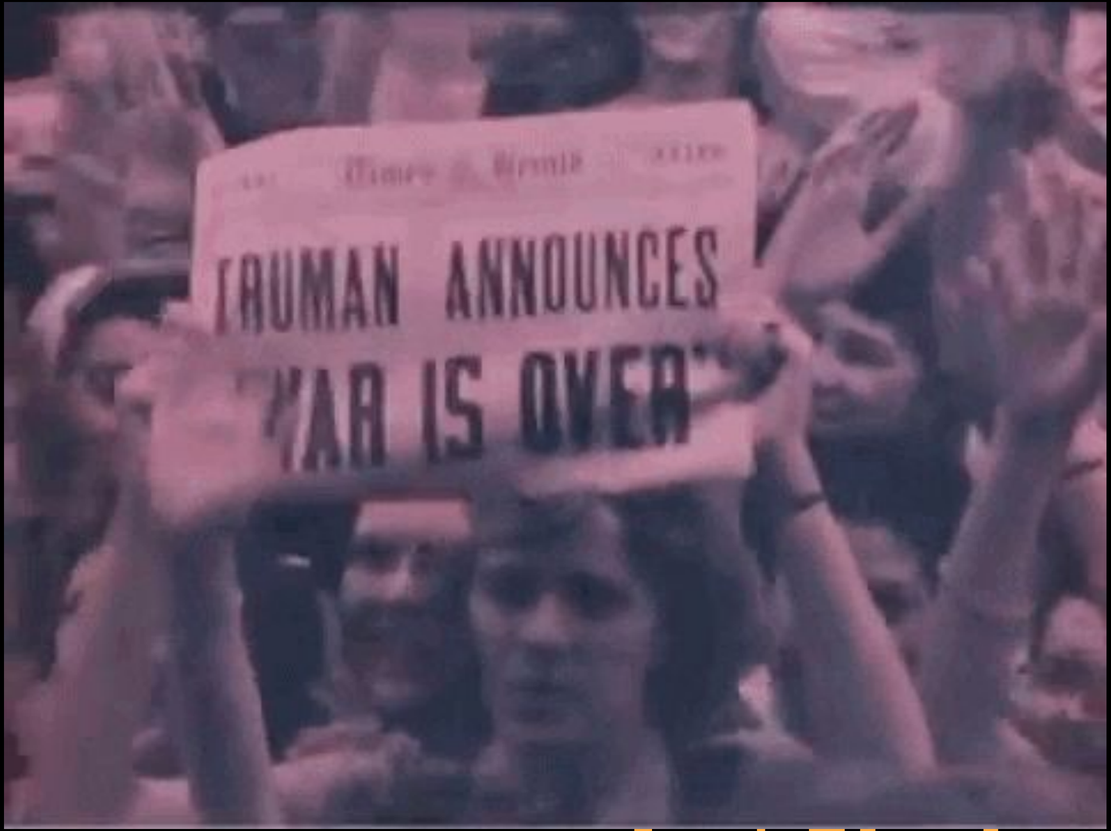
3. Serializations in other media types (defined by the VCWG) MUST be able to be transformed into the base media type.
    a. Another media type MUST identify if this transformation is one-directional or bi-directional.
    b. Bi-directional transformation MUST preserve @context.
    c. Transformation rules MUST be defined, but not necessarily by this WG.

Bonus Option: Pit of Death

04

What Now?

We're all on team Verifiable Credentials


Times-Herald
TRUMAN ANNOUNCES
"WAR IS OVER"

Break
(15 mins)

@context
(Chairs, 120 mins)

Lunch
(60 mins)

Verifiable Credentials

Updating Use Cases 2023

Short Use Cases

- Simple, relatable to all
  - e.g., Digital transcript
- Self-contained, unrelated to other use cases except in domain
  - e.g., Education domain, containing Digital transcript, Taking a test, Transferring schools, Online classes
- High level only
  - Title
  - Short paragraph
- Currently 30
  - Iterations and improvements welcome
  - Open to a few additions
  - More than that will require deletion

Extant Use Cases

- Illustrative of market adoption
  - Examples of use of W3 Verifiable Credentials in real-world deployments
- High level and Reference
  - Name
  - Single sentence description
  - URL
- New Section
  - Open to all W3C VC examples in the wild

Focal Use Cases

- Deeper dive on a few examples
- Detailed
  - Background
  - Distinction
  - Scenario
  - Parties (Issuer, Subject, Holder, Verifier)
  - Validation requirements
  - Relationships to or dependencies on other credentials
  - Example Artifacts (VCs, VPs)
  - Trust Hierarchy (Liabilities)
  - Threat Model (Risks & Responses)
- Currently 3 Focal Use Cases
  - Iterations and improvements welcome
  - Open to one or two additional

Timeline

1. Github (in process) — https://github.com/w3c/vc-use-cases/
   a. Switch to composition of separate files
   b. Add templates for PRs
2. Formal call for input
   a. Week of Mar 10 email
   b. Week of Mar 17 VCWG call
3. Special Topic / Use Case Calls
   a. April 14
   b. May 12
   c. June 9
4. Contribution Deadline – July 7
5. Draft – July 28
6. TPAC – September 11-15

Industry News
(Mike P, 60 mins)

Issue Processing (Chairs/Editors, 60 mins)

Break
(15 mins)

Issue Processing (Chairs/Editors, 30 mins)

Deliverables
(Chairs, 45 mins)

End of Day 3